A Novel Parameterization of the Trifocal Tensor

Joni De Guzman
UC San Diego
j5deguzm@eng.ucsd.edu

Anthony H. Thomas
UC San Diego
ahthomas@eng.ucsd.edu

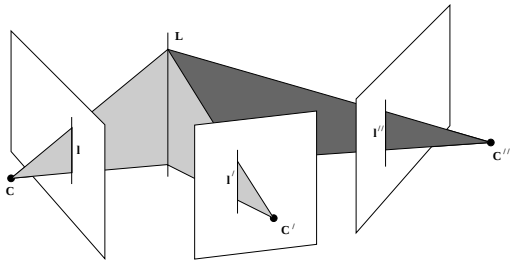
Abstract

The trifocal tensor encodes the geometric constraints present in three views of a single 3D scene and is an object of prime importance to imaging geometry. It is commonly applied to the problem of uncalibrated scene reconstruction where the additional constraints imposed by a third view yield significant improvements over two-view reconstruction. The theoretical properties of the tensor have been studied extensively, however only modest attention has been paid to techniques for its parameterization and estimation. Existing minimal parameterizations tend to be mathematically complex resulting in the use of inefficient non-minimal parameterizations. We propose a parameterization of the trifocal tensor based on a fundamental matrix and a projection matrix and demonstrate its use in a reconstruction pipeline.

1. Introduction

The intrinsic geometry defined by multiple corresponding images of a single 3D scene imposes constraints on the relationship between features across those images. The trifocal tensor embodies this geometric relationship between three corresponding images. It is the three view analog of the fundamental matrix, which relates features across two views. It is commonly applied to the problem of uncalibrated scene reconstruction. Throughout this work we assume the "pinhole" camera model in which a camera is described by a projection matrix $\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]$ consisting of a calibration matrix $\boldsymbol{K}$ containing the camera's internal parameters, an SO(3) rotation matrix $\boldsymbol{R}$, and a translation vector $\boldsymbol{t}$. The camera projects homogeneous points $\boldsymbol{X}$ in the real world, equivalently termed the "3D scene," into homogeneous points $\boldsymbol{x}$ in the image plane under this projection matrix according to: $\boldsymbol{x} = \boldsymbol{P}\boldsymbol{X}$. We here assume that the calibration matrix $\boldsymbol{K}$ is unknown, in which case the projection matrix can be described by: $\boldsymbol{P} = [\boldsymbol{M} \mid \boldsymbol{v}]$ where $\boldsymbol{K}$ has been subsumed into the rotation and translation components. We denote matrices and points in 3D with bold upper case letters. Vectors and points in the image plane are denoted with bold lower case letters. Scalars are denoted by standard lower case letters. We index views (i.e. images) by i where i ranges from 1 to 3 and data points by j where the range of j depends on the context. Points may be assumed to be homogeneous unless explicitly stated otherwise.

Figure 1. The trifocal tensor arises from the relationship between points and lines in three corresponding views.



2. Background on the Trifocal Tensor

The discussion below follows that of [5] chapter 15 which contains a fairly "feature-complete" discussion of the trifocal tensor. In the interests of concision, we omit some steps and discuss only a few salient properties of the tensor. Suppose there are three cameras imaging a 3D scene having projection matrices:

$$\boldsymbol{P}_1 = [\boldsymbol{I} \mid \boldsymbol{0}]$$
$$\boldsymbol{P}_2 = [\boldsymbol{A} \mid \boldsymbol{a}_4]$$
$$\boldsymbol{P}_3 = [\boldsymbol{B} \mid \boldsymbol{b}_4]$$

Then the trifocal tensor can be seen as $3 \times 3 \times 3$ ensemble of matrices $\{\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3\}$ where $\boldsymbol{T}_i = \boldsymbol{a}_i \boldsymbol{b}_4^T - \boldsymbol{a}_4 \boldsymbol{b}_i^T$, where $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ denote the i-th column of the projection matrices for cameras 2 and 3 respectively. From this form, two useful two useful facts emerge. First, the trifocal tensor may be computed explicitly from the projection matrices of three cameras imaging a 3D scene. Such a tensor is guaranteed to be geometrically valid and is said to be "consistent." Second, given a trifocal tensor it is possible to recover the underlying projection matrices and hence the fundamental

matrices. We will discuss the procedure for decomposing the tensor into its constituent projection and fundamental matrices in some detail in section 4.

We note as well that the choice of the first camera to be canonical was arbitrary. For any given 3D scene there are three valid trifocal tensors corresponding to the assumption that a different camera matrix is canonical. Each of the three tensors are distinct, but all are consistent with the 3D scene, and the following discussion is valid for whichever form is chosen.

Similar to the fundamental matrix, the trifocal tensor can be used to constrain the search for point correspondences across multiple views. Any pair of corresponding points $\boldsymbol{x}_1 \leftrightarrow \boldsymbol{x}_2$ in two views must satisfy the property $\boldsymbol{x}_2^T \boldsymbol{F} \boldsymbol{x}_1 = 0$. However, this a fairly loose constraint which will be satisfied for *any* $\boldsymbol{x}_2$ lying on the (epiploar) line defined by $\boldsymbol{l} = \boldsymbol{F}\boldsymbol{x}_1$. The primary virtue of the trifocal tensor is that it imposes a much stricter and more extensive set of constraints. For example, any triplet of corresponding points $\boldsymbol{x}_1 \leftrightarrow \boldsymbol{x}_2 \leftrightarrow \boldsymbol{x}_3$ must satisfy:

$$[\boldsymbol{x_2}]_x \left(\sum_i x_1^i \boldsymbol{T}_i\right) [\boldsymbol{x_3}]_x = \boldsymbol{0}$$

where $[\boldsymbol{x}]_x$ is the skew-symmetric representation of a three-vector $\boldsymbol{x}$ [5]. Similar constraints can be derived for other combinations of features across three images (i.e. point-line-line, line-line-line, etc...). These constraints are known as the trilinearities.

Although the trifocal tensor has 27 elements, it has only 18 degrees of freedom. To see why, note that the tensor can be constructed from three projection matrices, each of which has 11 degrees of freedom, yielding 33 degrees of freedom. However, the trifocal tensor is invariant to projective transformations of the 3D scene. This means that for a 4×4 matrix $\boldsymbol{H}$, the trifocal tensor corresponding to the projection matrices $\boldsymbol{P}_i$ and $\boldsymbol{P}_i\boldsymbol{H}$ are equivalent. Due to this invariance, the tensor looses 15 degrees of freedom yielding 18 degrees of freedom. This implies that any geometrically valid trifocal tensor must satisfy a set of eight internal constraints *á la* the rank 2 constraint on the fundamental matrix. However, the constraints on the tensor are considerably more complicated than the fundamental matrix, and it is generally preferable to ensure they are satisfied implicitly. [3] has an excellent discussion of the internal constraints on the trifocal tensor to which the interested reader is referred.

Estimation of the tensor from image data (described in [5] and [3]) is similar to many other projective entities and typically consists of obtaining a linear estimate followed by an iterative refinement, minimizing either algebraic or geometric error. Geometric error is generally considered the optimal criterion for minimization because it is equivalent to a maximum likelihood estimate under the assumption that noise follows a Gaussian distribution [5, 9]. The primary challenge in estimating the tensor is obtaining a suitable parameterization. In the following section we summarize the relevant literature on the trifocal tensor.

3. Related Work

The trifocal tensor is, in general, a well studied entity in the literature. Its use in scene reconstruction extends back at least to 1991 [13] although it was not then differentiated from higher order multi-view tensors (nor for that matter was it yet even described as a tensor) and its internal properties had not yet been formalized. [11] proposed the first minimal algorithm for its computation from six corresponding points. [2] and [15] showed that algorithms to compute the fundamental matrix from seven or more points in two views could be reformulated to compute the trifocal tensor by exchanging the algorithmic role of camera centers and points in the 3D scene. This discovery has come to be known as Carlsson-Weinshall duality and has important application to minimal scene reconstruction algorithms and n-view geometry.

Despite its importance to efficient estimation, the parameterization of the trifocal tensor has received only modest attention in the literature. [14] presents a minimal parameterization based on the six point algorithm of [11]. This parameterization maps a set of four points to the canonical projective basis and expresses the remaining two in this frame. However, while this parameterization is minimal, it is *not* one-to-one and results in three valid tensors which cannot be disambiguated without the use of a seventh point [3]. [4] derive a minimal parameterization based on the internal constraints of the tensor. However, this parameterization also results in three tensors which cannot be disambiguated without further information. The same authors subsequently derived a similar parameterization in [10] which is both minimal *and* one to one. A nontrivial complication with these parameterizations is that they are complex mathematically rendering them unapproachable for non-experts[1] without considerable study. Based partly on this logic, the canonical textbook in visual geometry [5] recommends a non-minimal parameterization which decomposes the tensor into two camera projection matrices (a third is assumed to be canonical) and simply applies bundle adjustment to the resulting twenty-two parameters. While this parameterization is straightforward conceptually, it is non-minimal and requires eight or more point correspondences compared to the six required under a minimal parameterization. Also noting the difficulty of existing minimal parameterizations, [9, 8] presents a minimal parameterization which uses three SO(3) matrices to transform a tensor such that ten well-defined elements are nonzero. The

[1] A category in which the authors of this paper firmly place themselves

resulting sparse tensor may then be parameterized using the three rotation matrices and the homogeneous ten vector of non-zero elements of the sparse tensor. This parameterization is arguably simpler than any of the previously known minimal parameterizations and results in only one unambiguous tensor. We here present an even simpler parameterization based on a fundamental matrix and a projection matrix. Fundamental matrices and projection matrices are the bread and butter of modern visual geometry and are widely taught concepts in computer vision. Our primary contributions in this work are:

- A novel parameterization of the trifocal tensor.
- A presentation of its use in a 3D reconstruction pipeline.
- Exposition of some practical issues in implementation.

4. Parameterizing the Tensor

We may parameterize the trifocal tensor using a fundamental matrix and a projection matrix[2]. The fundamental matrix is minimally parameterized using 7 parameters [1], and projection matrices may be minimally parameterized using 11 parameters [5]. The concatenation of these two parameter vectors yields the 18 requisite parameters needed to minimally parameterize the trifocal tensor. That the resulting tensor is consistent follows immediately from the fact that a fundamental matrix may be decomposed into two projection matrices where one is assumed to be canonical. Then, given the third projection matrix, it is possible to solve for a tensor of the form: $\mathcal{T}_i^{jk} = a_i^j b_4^k - a_4^k b_i^j$ which is the definition of a consistent trifocal tensor. Thus, there is no need to resort to complex proofs of consistency as with the case of other minimal parameterizations. The parametric form itself is a composition of well understood parameterizations. We now consider the technical details of this parameterization.

As was remarked in section 1, the trifocal tensor may be decomposed into two fundamental matrices $\boldsymbol{F}_{21}$ and $\boldsymbol{F}_{31}$ thereby characterizing the relationship between views I and II and I and III respectively. From either of these fundamental matrices it is possible to recover two projection matrices: $\boldsymbol{P}_1 \; assumedtobecanonical$ and $\boldsymbol{P}_2$ or $\boldsymbol{P}_3$ [5]. Without loss of generality, we arbitrarily choose to consider $\boldsymbol{F}_{21}$. This fundamental matrix captures the projective relationship between views I and II. We must then account for the third view. This is slightly more complex since the three views are not independent. [5] presents an intuitive demonstration of this dependence which proceeds as follows: from $\boldsymbol{F}_{21}$ we can reconstruct the 3D scene *up to a projective transformation*. Given a 3D scene and a known set of corresponding image points, it is a simple matter to solve for a consistent camera projection matrix. Therefore, the form of the third projection matrix depends on the projective frame defined by the first two in conjunction with $\mathcal{T}$. [5] goes on to provide a simple algorithm for computing the fundamental matrices:

1. Retrieve the epipoles $\boldsymbol{e}_2, \boldsymbol{e}_3$ according to the following:

   $$\boldsymbol{e}_2^T[\boldsymbol{u}_1 \mid \boldsymbol{u}_2 \mid \boldsymbol{u}_3] = \boldsymbol{0} \text{ and } \boldsymbol{e}_3^T[\boldsymbol{v}_1 \mid \boldsymbol{v}_2 \mid \boldsymbol{v}_3] = \boldsymbol{0}$$

   where $\boldsymbol{u}_i, \boldsymbol{v}_i$ are respectively the left and right null spaces of the $\boldsymbol{T}_i$ ("slices" of the tensor).

2. Recover the fundamental matrices as:

   $$\boldsymbol{F}_{21} = [\boldsymbol{e_2}]_x \boldsymbol{T}_i \boldsymbol{e}_3 \text{ and } \boldsymbol{F}_{31} = [\boldsymbol{e_3}]_x \boldsymbol{T}_i \boldsymbol{e}_2$$

   where $\boldsymbol{F}_{j1}$ denotes the "j-th" column of the appropriate fundamental matrix.

Given $\boldsymbol{F}_{21}$, there exists a straightforward algorithm (see [5] section 9.5) to recover the second projection matrix $\boldsymbol{P}_2$. Given $\boldsymbol{P}_2, \boldsymbol{e}_2, \boldsymbol{e}_3$ and $\mathcal{T}$, we can solve for $\boldsymbol{B}$ (the 3×3 rotation matrix of camera three). Let us assume that $\boldsymbol{a}_4^T \boldsymbol{a}_4 = 1$ (which is true if $\boldsymbol{a}_4$ is a unit vector) then:

$$\begin{aligned} \boldsymbol{T}_i &= \boldsymbol{a}_i \boldsymbol{b}_4^T - \boldsymbol{a}_4 \boldsymbol{b}_i^T \\ \Rightarrow \boldsymbol{b}_i &= \boldsymbol{a}_4^T (\boldsymbol{a}_i^T \boldsymbol{b}_4^T - \boldsymbol{T}_i^T) \end{aligned}$$

The translation vector $\boldsymbol{b}_4$ can be chosen to be $\frac{\boldsymbol{e}_3}{||\boldsymbol{e}_3||_2}$. However, this only determines $\boldsymbol{b}_4$ up to its sign. There are therefore two possible third projection matrices:

$$\boldsymbol{P}_3 = [\boldsymbol{B} \mid \boldsymbol{b}_4] \text{ and } \tilde{\boldsymbol{P}}_3 = [\boldsymbol{B} \mid -\boldsymbol{b}_4]$$

only one of which is consistent with the projective frame defined by $\boldsymbol{F}_{21}$. To ensure the correct projection matrix is chosen it is necessary to construct both and select the one which reconstructs to the correct tensor along with $\boldsymbol{P}_2$ under $\boldsymbol{T}_i = \boldsymbol{a}_i \boldsymbol{b}_4^T - \boldsymbol{a}_4 \boldsymbol{b}_i^T$.

To recapitulate, we demonstrated that the trifocal tensor could be parametrized by a decomposition into a fundamental matrix and a third projection matrix. The projection matrix must be chosen so as to be consistent with the projective frame defined by the fundamental matrix. We further noted that this parameterization is essentially its own proof of consistency.

5. Implementation

In the following sections, we provide additional detail on the algorithm for estimating the trifocal tensor using a 3D reconstruction pipeline as a motivating example. At a high

[2] We note that credit for this parameterization is due, in its entirety, to Ben Ochoa

level, our implementation follows the general procedure outlined in [5] chapter 16. Given a set of image point correspondences across three views, we first apply an outlier rejection algorithm to remove false matches. The set of image point correspondences is found by projecting a 3D point cloud onto three different image planes. Next, we initialize the trifocal tensor and 3D scene points via a linear estimate. Finally, we refine the result through nonlinear optimization which simultaneously solves for the optimal tensor and 3D scene. The minimal inputs to this pipeline are a triplet of image point correspondences $\boldsymbol{C} = \boldsymbol{x}_1 \leftrightarrow \boldsymbol{x}_2 \leftrightarrow \boldsymbol{x}_3$. The number of points in $\boldsymbol{C}$ is at least six. As discussed in section 1, this enables reconstruction only up to an overall projective ambiguity. This is sufficient for some purposes (i.e. performing point transfer), but is visually unsatisfying as the resulting 3D scene appears heavily distorted. We additionally allow the user to pass a set of five of more "control-points" with known 3D locations which can be used to upgrade the projective reconstruction to a metric one.

5.1. Outlier Rejection

Due to noise in the underlying image points, the putative feature point correspondences will likely contain false matches. Viewing the trifocal tensor as a generative "model" for the data, these false matches can be interpreted as outlier measurements. The issue of outlier rejection is again well studied and techniques generally fall into two categories: formulating estimators which are inherently robust to the presence of outliers or trimming outlier measurements prior to estimation. Following the recommendation of [5] we adopt the latter approach and use the M-sample consensus estimator (MSAC) to obtain a set of inlier correspondences.

A full treatment of the theory of consensus estimators is beyond the scope of this work and we here give only a cursory overview. The interested reader is referred to [5] which provides a thorough overview of the topic with applications to problems in geometry and vision. MSAC is an iterative algorithm which repeatedly estimates a model on a randomly selected subset of input data. At each iteration (i.e. for each model instance), the full set of observations is classified into inliers and outliers based on some metric of error (which will be discussed in more detail below). The model instance yielding the largest number of inliers is retained and the outlier points from that instantiation are trimmed from further analysis. The number of iterations and the threshold for outlier determination may be chosen such that bounds can be placed on the probability that at least one sample contains no outliers and the probability that an inlier is incorrectly rejected. MSAC may be "tuned" to be more or less permissive of outlier measurements depending on the constraints of the problem.

The chief issue in MSAC is obtaining a suitable procedure for efficiently estimating the model. The theory of MSAC relies on the use of a procedure for instantiating the model from the minimal number of data points. As noted above, the minimal solution to the trifocal tensor requires six point correspondences across three images. The minimal solution may be obtained by reconstructing the 3D scene generating the six sampled points and then applying a simple linear algorithm to obtain the corresponding camera matrices (from which the tensor can be computed) [11]. Reconstruction is typically performed by reformulating the problem in terms of its Carlsson-Weinshall dual which was described in section 3. However, a key step in dualization based algorithms is a mapping of the image points to the canonical projective basis for $\mathbb{P}^2$ which was shown in [12] not to preserve the shape of Gaussian distributions and lead to biased estimates of the resulting scene points. [12] proposes an algorithm which performs all computations in the original coordinate frame thus eliminating the source of bias and yielding a computationally simpler procedure which does not require a mapping to the canonical projective basis. This algorithm was shown to be algebraically identical to the method of [6] in the case of three view reconstruction. We therefore adopt the algorithm of [12] for obtaining the minimal solution.

Having obtained a minimal solution for the tensor, the remainder of the MSAC procedure is straightforward. For each triplet of points in the putative set of putative correspondences, we compute the error under the current model and classify those points with sufficiently low error as inliers. The precise nature of this error remains to be defined. Ideally, we would use the geometric error given by:

$$\epsilon_i = \sum_i d(\boldsymbol{x}_i^j, \boldsymbol{P}_i \hat{\boldsymbol{X}}^j)$$

where $j = 1, 2, 3$ and $\boldsymbol{P}_j$ is recovered from the estimated trifocal tensor according to the algorithm presented in section 4. However, this is a complex nonlinear problem which is expensive to estimate numerically. We instead use the Sampson error which is a first order approximation to geometric error and which has an analytic solution. The Sampson error is given by [5]:

$$\delta_s^j = \sum_j \epsilon_j^T (\boldsymbol{J}_j \boldsymbol{J}_j^T)^{-1} \epsilon_j$$

where $\boldsymbol{J}_j = \frac{\partial \epsilon_j}{\partial (\boldsymbol{x}_j^1, \boldsymbol{x}_j^2, \boldsymbol{x}_j^3)}$ and ϵ_j is the algebraic error given by one or more of the trilinear constraints discussed in section 1. Bellow we discuss the computation of ϵ in more detail.

Recall from section 1 that the trifocal tensor defines a set of constraints over feature correspondences across three views. Thus, the error for a given triplet of feature correspondences is the degree to which this triplet satisfies those

constraints. The simplest constraint to work with is the point-line-line constraint which may be formulated as (see [8]):

$$vect(\mathcal{T})^T \cdot (\boldsymbol{x}_1 \otimes \boldsymbol{l}_2 \otimes \boldsymbol{l}_3) = \epsilon$$

Where $vect(\mathcal{T})$ denotes the vectorization operator which reshapes the tensor into a 27×1 vector in row-major order and the error term $\epsilon = 0$ under a perfect correspondence. A minor complication is that the inputs to our algorithm are points, but the formulation above requires lines. It is simple to recover a pair of lines $\boldsymbol{l}^1$ and $\boldsymbol{l}^2$ passing through a point $\boldsymbol{x}$ as the left null space of that point. The equation for transfer-error for a given triplet of correspondences then becomes:

$$\boldsymbol{\epsilon_i} = \begin{pmatrix} \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^a \otimes \boldsymbol{l}_3^a \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^a \otimes \boldsymbol{l}_3^b \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^b \otimes \boldsymbol{l}_3^a \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^b \otimes \boldsymbol{l}_3^b \end{pmatrix} \cdot vect(\mathcal{T})$$

Once the Sampson error has been obtained the remainder of the algorithm is trivial. The observed point correspondences are classified into inlier and outliers based on a distance threshold, and the algorithm iterates until a sufficient number of samples have been gathered so as to meet the probabilistic guarantees described above.

5.2. Linear Estimate

To obtain an initial estimate of the tensor we follow the procedure outlined in algorithm 16.1 of [5]. This algorithm, commonly known as the "Direct Linear Transformation" (DLT), solves for the tensor as the solution to the homogeneous system of equations $\boldsymbol{At} = \boldsymbol{0}$, where $\boldsymbol{t}$ is a 27-vector containing the elements of the tensor. We now describe the method used to obtain the design matrix $\boldsymbol{A}$.

Observe that the equation $\boldsymbol{At} = \boldsymbol{0}$ will be satisfied if the rows of $\boldsymbol{A}$ embed any of the trilinearity constraints discussed in section 2. While any of the trilinearities may be used to construct $\boldsymbol{A}$ we found that the point-line-line constraint: $(\boldsymbol{x}_1 \otimes \boldsymbol{l}_2 \otimes \boldsymbol{l}_3)^T \cdot vect(\mathcal{T})$ was the easiest to use. Construction of $\boldsymbol{A}$ requires a minimum of 7 point correspondences and proceeds in a fashion identical to that described in 5.1:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^a \otimes \boldsymbol{l}_3^a \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^a \otimes \boldsymbol{l}_3^b \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^b \otimes \boldsymbol{l}_3^a \\ \boldsymbol{x}_1 \otimes \boldsymbol{l}_2^b \otimes \boldsymbol{l}_3^b \end{pmatrix}$$

The resulting system of equations can be solved efficiently using SVD. We found that using more than minimal number of point correspondences resulted in a more accurate linear estimate. Therefore, we use the entire set of data points to construct $\boldsymbol{A}$.

The resulting tensor may not be geometrically valid because it does not enforce the eight internal constraints. Therefore, we employ the method of algorithm 16.2 in [5] to satisfy the internal constraints. This method minimizes the algebraic error of the entries of the tensor. In summary, this algorithm involves retrieving the two epipoles from $\hat{\boldsymbol{T}}_i$ and minimizing $\|\boldsymbol{At}\|$ subject to $\|\boldsymbol{t}\| = 1$ where $\boldsymbol{A}$ is the same $4n \times 27$ matrix. Here, $\boldsymbol{t} = \boldsymbol{Ea}$ where $\boldsymbol{E}$ is a 27×18 matrix representation of $\boldsymbol{T}_i$ defined in 5.1 and $\boldsymbol{a}$ is the vector representation of tensor entries a_i^j and b_i^k.

Following generally established best practice we data normalize the input image points by computing transformations $\boldsymbol{H}_i$ which rescales the input points under $\tilde{\boldsymbol{x}}_i = \boldsymbol{H}_i \boldsymbol{x}_i$ such that their individual components are of mean zero, and the sum of the variances is $\sqrt{2}$. All estimation, linear and nonlinear, is performed on data normalized points and any subsequent expressions can be assumed to be data normalized. Estimation on normalized points yields "normalized" tensors which can be de-normalized according to [7]:

$$\boldsymbol{T}_i = \boldsymbol{H}_2^{-1} \sum_{i=1}^{3} (\boldsymbol{H}_1^T \tilde{\boldsymbol{T}}_i) \boldsymbol{H}_3^{-T}$$

Although the point is made in every paper addressing linear estimation of projective entities, we again stress that data normalization is essential to the stability of the DLT and the subsequent nonlinear optimization. Failure to data normalize can result in initial estimates that do not converge during nonlinear optimization or are heavily biased.

5.3. Nonlinear Optimization

The linear estimate obtained in 5.2 is suboptimal because it minimizes algebraic rather than geometric error. The final step of our algorithm consists of a nonlinear refinement to this preliminary estimate. In this step we minimize the geometric error function:

$$\epsilon = \sum_{i,j} d(\boldsymbol{x}_i^j, \boldsymbol{P}_i \boldsymbol{X}^j)^2$$

where $d(\cdot)$ is the euclidean distance, $\boldsymbol{x}_i^j$ are the true coordinates of the j-th point in the i-th image and $\boldsymbol{P}_i \boldsymbol{X}_j$ is estimated projection of the j-th scene point into the i-th image. Following the recommendation of [5] we employ the Levenberg-Marquardt (LM) algorithm to perform the minimization. For a thorough treatment of the LM algorithm the reader is once again referred to [5]. To provide the necessary context, we give a high-level *conceptual* overview of the LM algorithm applied to the problem of estimating the trifocal tensor in algorithm 1 which is based on appendix 6.2 of [5]. We stress that algorithm 1 is inefficient and should not be implemented verbatim. We now discuss some salient points of this algorithm.

5.3.1 Initialization of the Scene Points

Algorithm 1 simultaneously adjusts the parameters representing $\mathcal{T}$ and the scene points $\boldsymbol{X}$. $\mathcal{T}_0$ was initialized in 5.2 using the DLT. To obtain an initial estimate of $\boldsymbol{X}$ we apply a simple linear algorithm (see: [5]) which estimates $\boldsymbol{X}_j$ as:

$$\begin{pmatrix} \left[\boldsymbol{x}_1^j\right]^\perp \boldsymbol{P}_1 \\ \left[\boldsymbol{x}_2^j\right]^\perp \boldsymbol{P}_2 \\ \left[\boldsymbol{x}_3^j\right]^\perp \boldsymbol{P}_3 \end{pmatrix} \boldsymbol{X}_j = 0$$

where $[\boldsymbol{x}]^\perp$ denotes the left null-space of a 3-vector $\boldsymbol{x}$. We note that it is not necessary to resort to more complex optimal triangulation techniques. The linear algorithm is extremely simple to derive and implement and yields, for reasonable levels of noise, a good initial estimate.

5.3.2 Parameterizing the Fundamental Matrix

We employ a slightly modified version of the method proposed by [1] to obtain a parametric representation of $\boldsymbol{F}_{21}$ based on its singular value decomposition. We summarize this parameterization here. Let $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}$ be a singular value decomposition of the fundamental matrix. Then $\boldsymbol{U}$ and $\boldsymbol{V}$ are O(3) matrices which can be upgraded to SO(3) matrices (i.e. rotation matrices) under simple transformation (multiplying the third column by -1 if $\det(\boldsymbol{U}) < 0$). There are several parameterizations of rotation matrices using Euler angles, quaternions or matrix exponentials. Following [5] appendix A4.3 we adopt the "angle-axis" parameterization based on matrix exponentials:

$$[\boldsymbol{\omega_u}]_x = \log(\boldsymbol{U})$$
$$[\boldsymbol{\omega_v}]_x = \log(\boldsymbol{V})$$

where $\boldsymbol{\omega}_u, \boldsymbol{\omega}_v$ are 3-vectors containing the parameters of $\boldsymbol{U}$ and $\boldsymbol{V}$ respectively. The parametric representation of $\boldsymbol{U}$ and $\boldsymbol{V}$ yields the first six parameters. The seventh parameter comes from the singular values: $\boldsymbol{\sigma} = diag(\Sigma)$. Because $rank(\boldsymbol{F}) = 2$, $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, 0)^T$. [1] applied a rescaling of Σ such that $\boldsymbol{\sigma} = (1, \sigma, 0)^T$ to obtain the seventh parameter. We deviate from [1] slightly by instead parameterizing $\boldsymbol{\sigma}$ according to the parameterization of the n-sphere described in [5] appendix A6.9.3. This parameterization constrains $\boldsymbol{\sigma}$ such that $||\boldsymbol{\sigma}|| = 1$ and is advantageous because it preserves the signs of the σ which we found to be important in practice. Thus, our parametric representation of the fundamental matrix $\boldsymbol{F}_{21}$ is a 7-vector: $(\hat{\boldsymbol{\omega}}_u, \hat{\boldsymbol{\omega}}_v, \hat{s})^T$.

5.3.3 Parameterizing the Projection Matrix

We parameterize the third projection matrix as a homogeneous 12-vector following the method of [5] appendix A6.9.2. As described there, a homogeneous vector $\boldsymbol{v}$ may be parameterized according to $\frac{2}{sinc(cos^{-1}(a))}$ where $a = cos(\frac{||\boldsymbol{v}||}{2})$. Note that the projection matrix is only determined up to its sign under this parameterization. This is not problematic as projection matrices are only determined up to scale. The key requirement on the third projection matrix is that it be consistent with the projective frame of $\boldsymbol{F}_{21}$. Thus, the parametric representation of $\boldsymbol{P}_3$ is an 11-vector: $(\hat{p}_1, ..., \hat{p}_{11})^T$. Observe that this parameterization reduces the dimension of the input vector by 1.

5.3.4 Parameterizing the 3D Scene

Each scene point is represented as a homogeneous 4-vector. We parameterize the 3D scene points using the same parameterization of homogeneous vectors as was employed in 5.3.3 for the projection matrix. Therefore, the parametric representation of the 3D scene is the concatenation of N 3-vectors of the form $(\hat{X}_j^1, \hat{X}_j^2, \hat{X}_j^3)^T$, yielding a $3N$ parameter vector: $(\hat{\boldsymbol{X}}_1, ..., \hat{\boldsymbol{X}}_N)^T$.

5.3.5 Sparse Jacobian

The parametric representation of the trifocal tensor is the concatenation of the components described in the previous three sections yielding a $3N + 18$ element vector of the form:

$$\hat{\mathcal{T}} = (\hat{\boldsymbol{\omega}}_u, \hat{\boldsymbol{\omega}}_{\boldsymbol{v}}, \hat{s}, \hat{p}_1, \cdots \hat{p}_{11}, \hat{\boldsymbol{X}}_1, \cdots \hat{\boldsymbol{X}}_N)^T$$

The Jacobian is then the derivative of this parameter vector with respect to the measurement vector. The measurement vector consists of the concatenation of the N corresponding inhomogeneous image points across the three views to yield a $6N$ element vector of the form:

$$\boldsymbol{x}_1 = x_1^1, y_1^1, x_1^2, y_1^2, ..., x_1^N, y_1^N$$
$$\boldsymbol{x}_2 = x_2^1, y_2^1, x_2^2, y_2^2, ..., x_2^N, y_2^N$$
$$\boldsymbol{x}_3 = x_3^1, y_3^1, x_3^2, y_3^2, ..., x_3^N, y_3^N$$
$$\Rightarrow (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3)^T$$

The resulting Jacobian follows a block structure where many of the entries are zero:

$$J = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_1 \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{B}_2 \\ \boldsymbol{0} & \boldsymbol{C} & \boldsymbol{B}_3 \end{pmatrix}$$

Here

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}^1 \\ \boldsymbol{A}^2 \\ \cdots \\ \boldsymbol{A}^N \end{pmatrix} \quad \text{where} \quad \boldsymbol{A}^j = \frac{\partial \boldsymbol{x}_2^j}{\partial(\hat{\boldsymbol{\omega}}_u, \hat{\boldsymbol{\omega}}_v, \hat{s})}$$

and

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{C}^1 \\ \boldsymbol{C}^2 \\ \cdots \\ \boldsymbol{C}^N \end{pmatrix} \quad \text{where} \quad \boldsymbol{C}^j = \frac{\partial \boldsymbol{x}_3^j}{\partial \hat{\boldsymbol{p}}} \quad .$$

The $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, and $\boldsymbol{B}_3$ blocks are themselves sparse block-diagonal matrices of the form:

$$\boldsymbol{B}_i = \begin{pmatrix} \boldsymbol{B}_i^1 & & & \\ & \boldsymbol{B}_i^2 & & \\ & & \ddots & \\ & & & \boldsymbol{B}_i^N \end{pmatrix} \quad \text{where} \quad \boldsymbol{B}_i^j = \frac{\partial \hat{\boldsymbol{x}}_i^j}{\partial \hat{\boldsymbol{X}}^j}$$

This sparse Jacobian is exploited to optimize the solution to the weighted least squares problem $\delta = (\boldsymbol{J}^T \Sigma_x^{-1} \boldsymbol{J} + \lambda \boldsymbol{I})^{-1} (\boldsymbol{J}^T \boldsymbol{\epsilon})$ which forms the backbone of LM. The δ can be found by following the general sparse LM algorithm A6.4 in [5], which involves creation of a normal equations matrix and normal equations vector.

6. Experimental Evaluation

In this section we present results from a preliminary investigation into the convergence and stability properties of our algorithm. We compare the rate of convergence and the accuracy of the resulting tensor. To test our algorithm we synthesize images of a publicly available point cloud[3] showing the interior of an office building. We then fix a projection matrix to be canonical and generate a pair of synthetic cameras by rotating and translating this first camera. The 3D scene is projected into the image planes and zero-mean Gaussian noise is added. We reject any points that do not lie in front of the cameras.

To assess our algorithm, we vary both the amount of noise and the number of points. We terminate the algorithm on two conditions: 1 it fails to converge after 1,000 iterations or 2 the difference in cost between any successive pair of iterations is less than 10^{-4}. All estimation is performed on data normalized points. For each reported value of noise and number of points, we repeat the experiment thirty times and report the median value. Table 1 reports the number of iterations required for the algorithm to converge while varying the number of observations and the standard deviation of the added noise. Table 2 reports the sum of squared errors between the true and estimated epipoles of the tensor after nonlinear refinement while varying the same parameters as Table 1.

For the scene reconstruction, we estimate the trifocal tensor from a sample of 300 points and add noise with standard deviation 0.01. We then reconstruct the 3D scene via triangulation using the MLE estimates for each camera projection matrix. Finally, we elevate the projective reconstruction to a metric one by using the known 3D location of a set of points to estimate the 3D planar projective transformation that maps the reconstructed scene from projective to euclidean space. Figure 2 shows the original point cloud after rejecting points not lying in front of each camera. Figure 3 shows the reconstruction of the original scene using our pipeline and the estimation of the trifocal tensor from 300 sampled points and 0.01 added Gaussian noise.

In general, the location of a set of 3D points will not be known, and more complex methods must be employed to address projective ambiguity. The mean squared error between the ground truth and reconstructed points is 0.0183.

[3] http://www.digital210king.org/downloader.php?file=10

Algorithm 1: Nonlinear Estimation of $\mathcal{T}$

Data: $\mathcal{T}_0$ - a linear estimate of the trifocal tensor. $\boldsymbol{C}$ - a set of feature correspondences. Σ_x - the covariance matrix of $\boldsymbol{C}$

Set $\boldsymbol{P}_1 \leftarrow [\boldsymbol{I}_{3\times3} \mid \boldsymbol{0}_{3\times1}]$
Extract $\boldsymbol{F}_{21}$ from $\mathcal{T}_0$
Extract $\boldsymbol{P}_2$ from $\boldsymbol{F}_{21}$
Extract $\boldsymbol{P}_3$ from $\mathcal{T}_0$ such that it is consistent with $\boldsymbol{P}_2$
Initialize an estimate $\boldsymbol{X}$ of the 3D scene

$\hat{\boldsymbol{p}} \leftarrow parameterize(\boldsymbol{P}_3)$
$\hat{\boldsymbol{f}} \leftarrow parameterize(\boldsymbol{F}_{21})$
$\hat{\boldsymbol{x}} \leftarrow parameterize(\boldsymbol{X}_0)$
$\hat{\boldsymbol{\theta}} \leftarrow [\hat{\boldsymbol{p}}, \hat{\boldsymbol{f}}]$

$\boldsymbol{\epsilon}_0 \leftarrow \sum_{i=1}^{3} d(\boldsymbol{x}_i^j, \boldsymbol{P}_i \boldsymbol{X}_j) \forall j$
$\lambda \leftarrow 10^{-3}$
while *not converged* **do**
  $\boldsymbol{J} = \frac{\partial \boldsymbol{C}}{\partial \hat{\boldsymbol{\theta}}}$
  $\delta = (\boldsymbol{J}^T \Sigma_x^{-1} \boldsymbol{J} + \lambda \boldsymbol{I})^{-1} (\boldsymbol{J}^T \Sigma_x^{-1} \boldsymbol{\epsilon})$
  $\hat{\boldsymbol{\theta}} \leftarrow \hat{\boldsymbol{\theta}} + \delta$

  $\boldsymbol{P}_2, \boldsymbol{P}_3, \boldsymbol{X} \leftarrow deparameterize(\hat{\boldsymbol{\theta}})$
  $\boldsymbol{\epsilon}_1 \leftarrow \sum_{i=1}^{3} d(\boldsymbol{x}_i^j, \boldsymbol{P}_i \boldsymbol{X}_j) \forall j$
  $\zeta \leftarrow \boldsymbol{\epsilon}_1^T \boldsymbol{\epsilon}_1 - \boldsymbol{\epsilon}_0^T \boldsymbol{\epsilon}_0$
  if $\zeta > 0$ **then**
    $\lambda \leftarrow 10\lambda$
    goto 13
  end
  if $|\zeta| < \tau$ **then**
    converged $\leftarrow$ true
  end
  $\lambda \leftarrow 10^{-1}\lambda$
  $\epsilon_0 \leftarrow \epsilon_1$
end

Figure 2. The Original 3D Scene (Ground Truth)

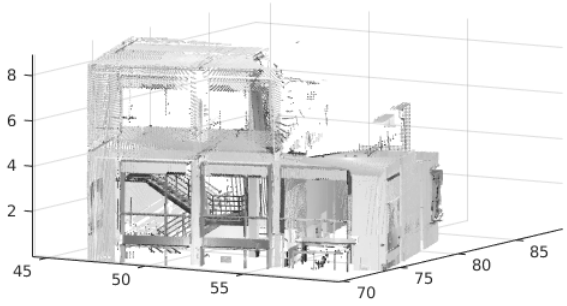
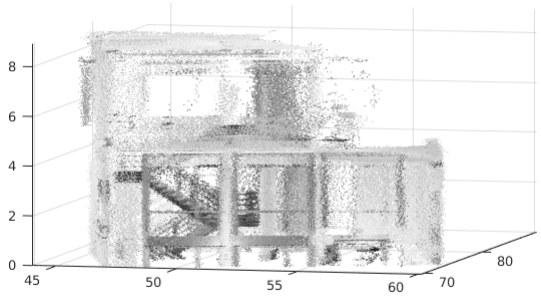


Figure 3. Reconstructed Scene



Notes: Figure 3 shows the 3D scene reconstruction using the camera projection matrices estimated through the LM algorithm.

We find that the results are in line with our expectations. Unsurprisingly, increasing the amount of added noise to the input points generally yields a poorer solution. Interestingly, using additive noise with $\sigma = 0.01$ seems to yield a better solution than $\sigma = 0.005$ for all numbers of points. We make the following two comments about these experimental results. First, there is at least one bug in our current implementation which causes the algorithm to initialize to an extremely high cost for certain subsets of points. Second, for reasonably sized samples of points (< 500) this initial extremely high cost occurs infrequently but can cause the algorithm to fail to converge.

Table 1. Median Iterations to Convergence

	Number of Observations		
σ	7	100	300
.005	9.0000	5.0000	5.0000
.01	6.5000	4.5000	7.0000
.05	11.0000	10.0000	8.5000

Notes: Table shows the median number of iterations required for convergence in the LM algorithm. Note that the number of iterations required for convergence is generally increasing with the amount of noise but is fairly stable with the number of points.

Table 2. Median Error in the ML-Estimate

	Number of Observations		
σ	7	100	300
.005	0.0911	0.0235	0.0099
.01	0.0622	0.0109	0.0166
.05	3.9866	0.0645	0.0519

Notes: Table shows the median sum-squared-error between the true and actual epipoles after nonlinear optimization. Note that error is generally increasing in noise, but decreasing in the number of points.

7. Future Work

We plan to extend our current work in two main directions. First, we plan to extensively evaluate the empirical properties of our parameterization and compare it against other known minimal parameterizations as well as to the non-minimal "gold-standard" parameterization of projection matrices. The minimal parameterizations have a propensity to get stuck in local minima during optimization [1] or to exhibit poor convergence rates. Therefore, we believe it is important to more completely understand how these issues effect our parameterization.

Second, we plan to explore the use of similar techniques to parameterize higher order multi-view tensors. In principle our procedure could be extended to these tensors by incorporating additional fundamental matrices. However, we note that these tensors rarely arise in practice and the issue of their minimal parametrization is largely one of interest rather than practicality.

8. Conclusion

We presented a novel technique for parameterizing the trifocal tensor based on a decomposition into a fundamental matrix and a camera projection matrix. We noted that this parameterization was guaranteed to yield a consistent trifocal tensor as a fundamental matrix and projection matrix can, together, characterize the geometry of three views. Additionally, we demonstrated the use of this parameterization in a 3D reconstruction pipeline and highlighted some important practical issues in its estimation. Our hope is that a

simpler minimal parameterization will encourage the use of the trifocal tensor in uncalibrated scene reconstruction over its less advantageous two view analog.

9. Acknowledgements

We are grateful to Ben Ochoa for providing the idea for the parameterization we have described and for many helpful discussions on geometry and the gory details of implementing our algorithm.

References

[1] A. Bartoli and P. Sturm. Nonlinear estimation of the fundamental matrix with minimal parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3):426–432, 2004. 3, 6, 8

[2] S. Carlsson. Duality of reconstruction and positioning from projective views. In *Representation of Visual Scenes, 1995.(In Conjuction with ICCV'95), Proceedings IEEE Workshop on*, pages 85–92. IEEE, 1995. 2

[3] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. MIT press, 2004. 2

[4] O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of 3 views. In *Computer Vision, 1998. Sixth International Conference on*, pages 477–484. IEEE, 1998. 2

[5] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 1, 2, 3, 4, 5, 6, 7

[6] R. I. Hartley. Projective reconstruction and invariants from multiple images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1036–1041, 1994. 4

[7] J. Li. *The Trifocal Tensor and Its Applications in Augmented Reality*. PhD thesis, University of Ottawa, 2005. 5

[8] K. Nordberg. A minimal parameterization of the trifocal tensor. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1224–1230. IEEE, 2009. 2, 5

[9] K. Nordberg. The key to three-view geometry. *International journal of computer vision*, 94(3):282–294, 2011. 2

[10] T. Papadopoulo and O. Faugeras. A new characterization of the trifocal tensor. *Computer VisionECCV'98*, pages 109–123, 1998. 2

[11] L. Quan. Invariants of six points and projective reconstruction from three uncalibrated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):34–46, 1995. 2, 4

[12] F. Schaffalitzky, A. Zisserman, R. Hartley, and P. Torr. A six point solution for structure and motion. *Computer VisionECCV 2000*, pages 632–648, 2000. 4

[13] M. Spetsakis and J. Y. Aloimonos. A multi-frame approach to visual motion perception. *International Journal of Computer Vision*, 6(3):245–255, 1991. 2

[14] P. H. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(8):591–605, 1997. 2

[15] D. Weinshall, M. Werman, and A. Shashua. Duality of multipoint and multi-frame geometry: Fundamental shape matrices and tensors. *Computer VisionECCV'96*, pages 217–227, 1996. 2